YAHOO! MAIL Classic

CHIPS Newsletter Vol 16 Sunday, June 12, 2011 9:45 AM

From: "chips@elproducts.net" <chips@elproducts.net>
To: "Chuck Hellebuyck" <chuck@elproducts.com>

CHIPs newsletter

Vol 16 / June 2011

Official Newsletter of
Chuck Hellebuyck's Electronic Products

http://www.elproducts.com/

Topic

Maximite Project
Creating my own Maximite
Conclusion

Maximite Project

I've been asked in feedback from readers to consider doing projects for the newsletter and decided to try that out in this newsletter. I decided to attempt to recreate a computer project I recently discovered called the Maximite by Geoff Graham. Maximite is an open sourced PIC32 based computer that recreates the old Radio Shack TRS80 and Commodore computer days. Only rather than build the actual Maximite I decided to create a simplified version on a PIC32 board to test out the concept. That is this months newsletter project.

MAXIMITE Computer

This project was interesting to me because my first computer was a Commodore

VIC20 that had a BASIC interpreter preprogrammed into the 6502 Microprocessor. It only had 5k of RAM memory for programs so compared to today's little PIC® Microcontrollers that is not much space but people still did some amazing things with these computers. I would occasionally drop by my local Radio Shack and play with their TRS80 computer. It had a lot more memory and many more features. All these early computers shared a common programming language call BASIC and though there were some differences, if you knew how to program one you could program the other.

Commodore VIC20

You had to connect the VIC20 to a television as that was your monitor. The TRS80 had its own composite monitor. Your programs were stored in RAM as you wrote them so if you lost power you lost your program. To save a program, you stored it on a tape drive which used audio cassette tapes. Eventually external floppy drives were released but very expensive. People wrote games and programs in BASIC, stored them on cassette tapes and then sold the tapes the same way video game disks are sold today.

TRS 80

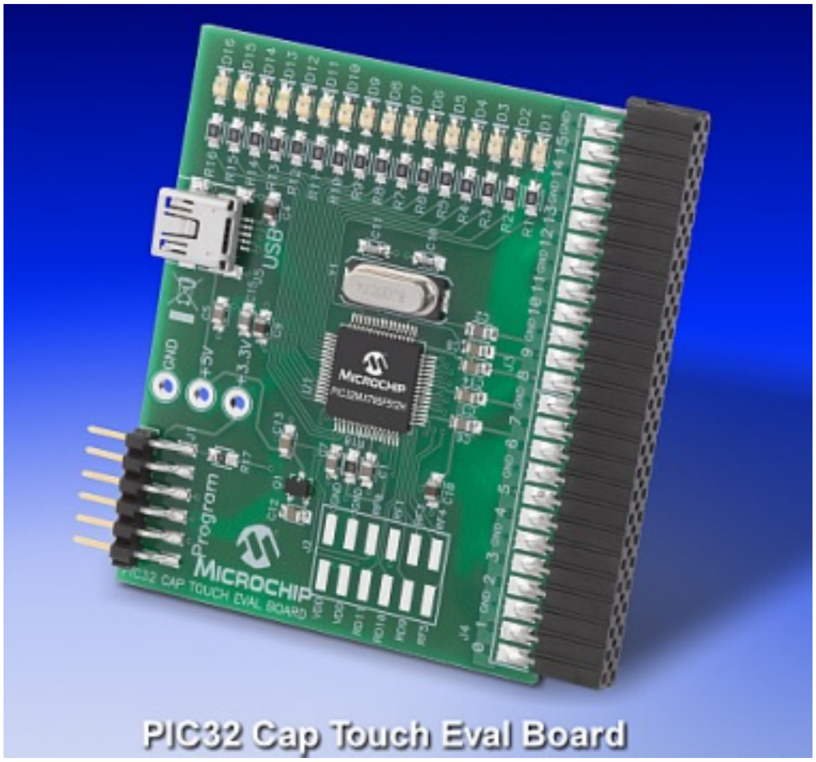
Now jump to the present. I have often thought about making a BASIC interpreter for a PIC using the PICBASIC PRO compiler as the develop software but never really found the time to do it. While reading an electronics forum I came across a reference to this new computer project called the Maximite which was developed by Geoff Graham in Australia. He wanted to recreate one of these older machines using a PIC32 so he went about writing his own BASIC interpreter and created the hardware around it. He called it the Maximite and included a VGA port for driving a monitor and a keyboard port for connecting to a keyboard. Add power to the power port and you have a miniature TRS80 style computer. He even added an SD card slot so you can save your programs.

In addition to all this he also put a USB port on the board which allows you to bootload any updates to the PIC32. He also programmed the Maximite to send and receive data from the USB port. This allows you to write BASIC programs on the Maximite with a terminal program without the need to hook up a monitor or keyboard. He also included an expansion port that has analog and digital pins from the PIC32 brought out to a 20 pin header connector. Within his BASIC interpreter he included commands to control these digital and analog ports. He released the schematic, board layout and the entire code as open source. He even showed a fundamental schematic on his website that shows the minimum connections to get the Maximite running. This is where my project begins.

Creating my own Maximite

After looking over the simplified schematic, I realized there was probably a development board from Microchip® with a PIC32 on it that might, with a little modification, allow me to test the Maximite software. Was I surprised to find that the PIC32 mTouchTM board from Microchip had the exact same PIC32MX795F512H micro installed with all the minimum requirements his simplified schematic required including the USB connection. Even some of the I/O he used for his analog/digital control ports were connected to either the mTouch boards expansion header or to prewired LEDs. This setup, I thought, would allow me to program the Maximite code into the PIC32 with a PICkit 3 and then program it in BASIC from a terminal program via the USB connection. And I could do this without bringing out the soldering iron. To test it out I decided to dust off my old VIC20 BASIC programming skills and try to create a simple program to create a train crossing type LED flasher with two output pins connected to LEDs. The first step was to get a hold of this board and check it out. All the PIC32 mTouch board schematics are available inside the users manual and that is free to download from the Microchip website.

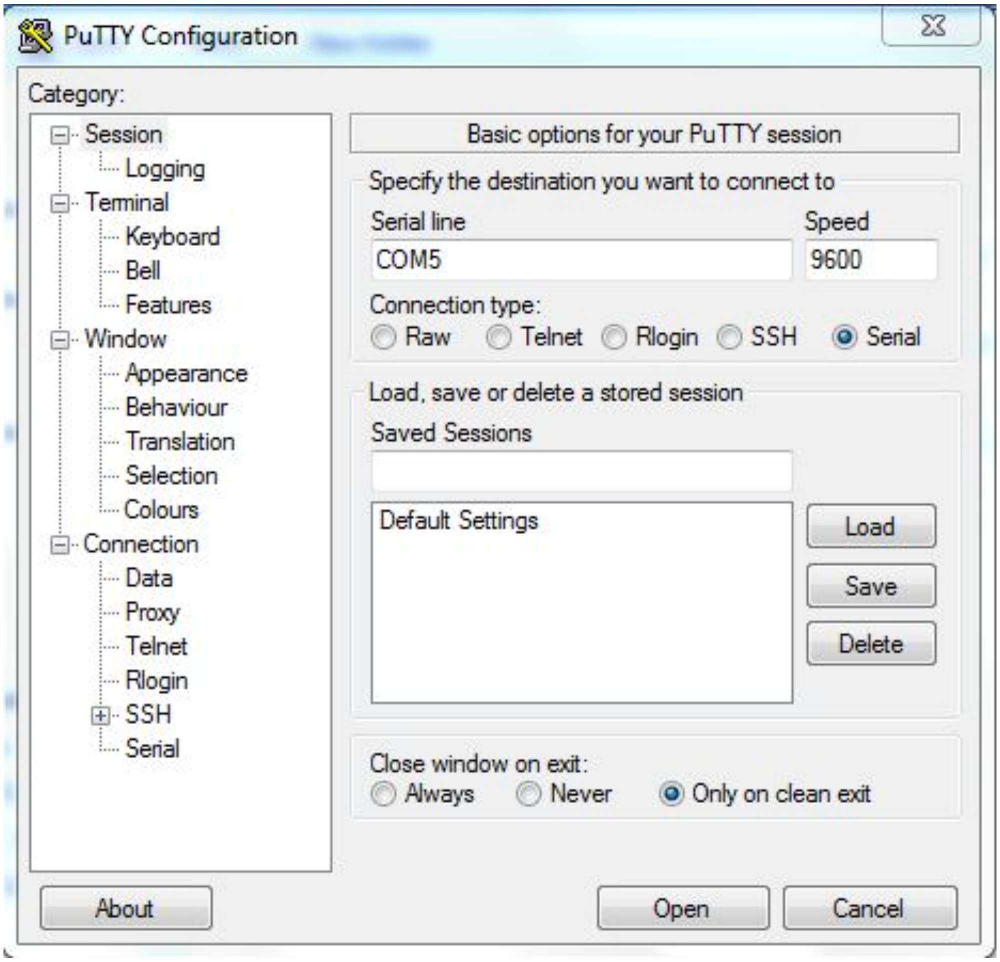
ThePIC32 mTouch board is shown below and sells for $24.95 under the part number AC323026.



PIC32 mTouch Board

The board already has the PICkitTM programming header so I plugged in my PICkit 3 and downloaded the software from Geoff's website http://geoffg.net/maximite.html. I used MPLAB to rebuild it using the Microchip C32 compiler (which comes in a free to download lite version) and then programmed the module to make it a Maximite. I wasn't done though. I had to install a serial port driver for the USB communication on my PC so the module will look like a serial COM port. This is also available to download from Geoff's site but for Windows computers only.

I installed this driver on my laptop running Windows 7. Windows 7 doesn't have Hyperterminal so I also downloaded the PuTTY terminal program that Geoff used in the description on his website. You can get Putty from the following website: http://www.chiark.greenend.org.uk/~sgtatham/putty. After installing PuTTY and unzipping the driver files, I then connected the PIC32 board to the computer's USB port and was prompted to select the driver. I ran the install wizard in Windows just like any new USB device and directed it to where I installed the driver. The process went smooth and it showed up as COM5 in a pop up screen after the driver installation was complete.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM5

Speed: 9600

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

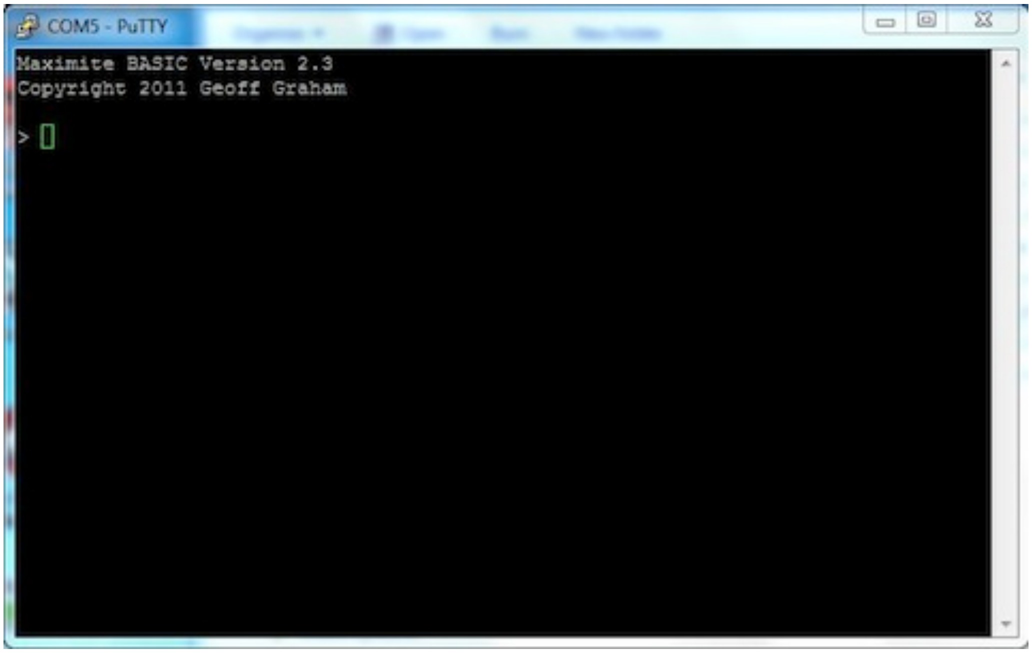
Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY Setup Screen

Then I started PuTTY and the configuration screen showed up. I selected the connection type "serial" and then typed in COM5 to the Serial Line box. I clicked open and the PuTTY screen appeared but blank. I thought maybe I did something wrong or maybe this board did need some modification. Then I hit the return key and the Maximite screen came to life as seen below.



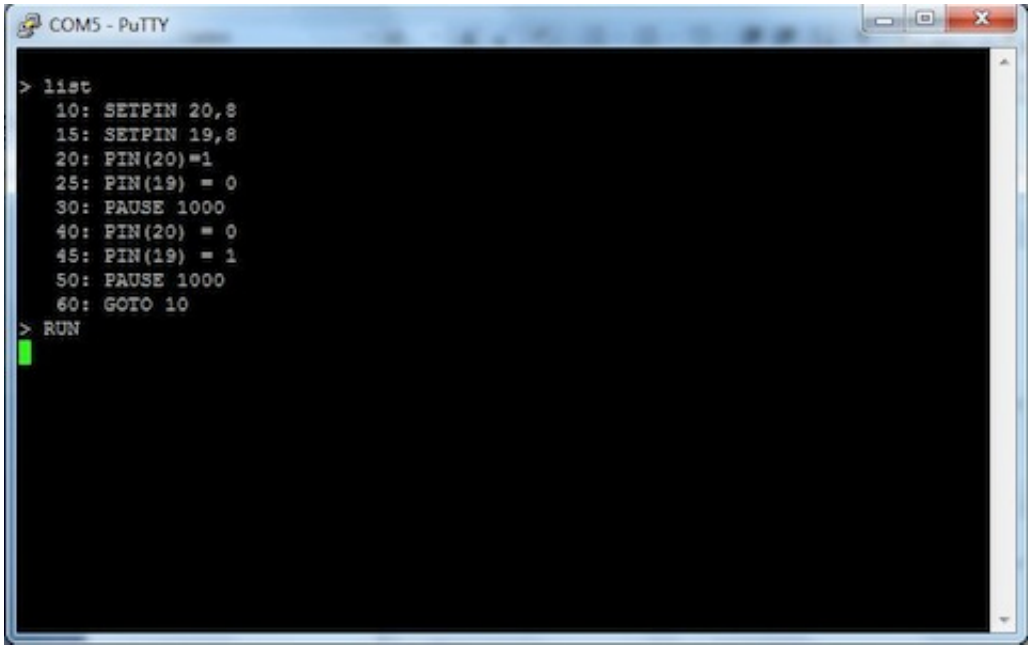
```
Maximite BASIC Version 2.3
Copyright 2011 Geoff Graham

>
```

Maximite Start up Screen

From there I had to recall my old days of programming in VIC20 BASIC and write a quick program to flash two LEDs in a back and forth motion just like a train crossing. I had to refer to Geoff's software documentation a couple times and read through the I/O control commands and also discovered a pause command as well. The VIC20 and TRS80 didn't have commands for controlling I/O ports directly so this was a new feature for this type of programming. Back then to control the VIC20's expansion port (which was the external I/O for that computer) you had to use the PEEK and POKE commands to control the data registers in the 6502 micro. This new Maximite software was beginning to look more like the PICBASIC PRO language I wrote my first book about.

The last two LEDs on the PIC32 mTouch board were pins 19 and 20 in Geoff's Maximite software. The program below was entered one line at a time. Each line needed its own unique line number. If I made a mistake I had to retype the whole line including the line number. Man did this bring back memories. Because each character is sent to the Maximite serially, I couldn't just backspace and retype. That is the way I had to write programs on the VIC20. If I had a syntax error, the Maximite sent back an error message and I had to retype it again.



```
> list
   10: SETPIN 20,8
   15: SETPIN 19,8
   20: PIN(20)=1
   25: PIN(19) = 0
   30: PAUSE 1000
   40: PIN(20) = 0
   45: PIN(19) = 1
   50: PAUSE 1000
   60: GOTO 10
> RUN
```

Train Crossing Program

Once the program was entered I typed the "LIST" command and it displayed the whole program. Now the moment of truth. I typed "RUN" at the command prompt and success; the LEDs began to flash back and forth. I tried different pause delays and reprogrammed it many times. It worked perfectly. I could now test out the Maximite without ever touching a soldering iron. I cannot explain what a thrill this was for me. It took me back to the days of my VIC20. It made me remember the first time I successfully controlled the expansion port on my VIC20.

After I had learned a lot about the VIC20, I purchased a VIC20 programmer's reference manual which had the schematic and all the inner details about the VIC20's internal setup. With this information I figured out how to control the expansion port on the VIC20 using the PEEK and POKE commands in Commodore BASIC language and had connected an old SPO256 speech chip with amplifier to the expansion port. This was a chip Radio Shack sold that allowed you to create speech with the proper control of a few pins on the chip. Anyway, it wasn't long before I had the VIC20 talking with a computer robot voice. Anything I typed was converted to the proper signals and spoken by this chip. Just getting those LEDs to flash proved this type of project was possible with the Maximite though with all the advanced capabilities of the compilers for Microchip PICs I'm not sure this has the same appeal. The Maximite can draw bitmap graphics and do floating point math so recreating the old games is where most of the applications for this seem to be focused.

My next step is to build an actual Maximite but the only resellers are in Australia and they are sold out. Maybe I'll make my own when I have more time. All the board files and parts list are open sourced so anybody can build their own.

Conclusion

I had so much fun with this I had to share it here with my newsletter readers. In response to last months request for any information on Microchip PIC based Arduino style modules, I received information about a 40 pin Arduino module I'll probably write about in next month's letter. If you know of any other PIC based Arduino style modules please send the info on so I can include them as well. If you develop something interesting like Maximite, send me an email and I might post it in a future newsletter. In fact email me any feedback, good or bad to chuck@elproducts.com.

If you get this newsletter forwarded to you then please sign-up for any future newsletters at my website www.elproduct.com. It's free.

Note: The Microchip name and logo, PIC are registered trademarks of Microchip Technology Inc. in the USA and other countries. Any other trademarks mentioned herein are the property of their respective companies.

Unsubscribe from CHIPs Newsletter